

IMPORTANTE:

- Crear un proyecto con su Apellido y nombre.
- Añadir comentarios identificando a qué ejercicio e inciso corresponde cada función. No alcanza con comentar en el Main o en los prototipados, **EN CADA FUNCIÓN DEBE COMENTARSE A QUÉ EJERCICIO E INCISO CORRESPONDE. NO SE CORREGIRÁ CÓDIGO NO IDENTIFICADO.**

Una empresa de venta de indumentaria deportiva nos ha encomendado realizar un pequeño sistema para manejar el stock de productos en sus distintas sucursales.

La información se encuentra almacenada en el archivo subido al campus, el cual hay que descargar y copiar dentro de la carpeta del proyecto del examen. Se trata de un archivo de estructuras "stRegistro" que responde a la siguiente estructura de datos:

```
typedef struct {
    int idSucursal;
    char nombreSucursal[25];
    char nombreProducto[25];
    char deporte[25];
    int stockProducto;
}stRegistro;
```

Se nos pide desarrollar la siguiente funcionalidad (**a los fines de facilitar la corrección, desarrollar todas las funciones en el archivo main.c en lugar de hacerlo en librerías:**):

1. Codificar las estructuras y funciones que hagan falta para **recorrer el archivo** adjuntado y **cargar los datos en una FILA** que implemente una **Lista Doblemente Enlazada**.
Modularizar respetando la responsabilidad de cada TDA.

20 puntos

2. **Mostrar la FILA** de forma **RECURSIVA**.
Modularizar respetando la responsabilidad de cada TDA.

A los fines de facilitar la tarea, se entrega el código correspondiente al mostrado de la estructura:

```
puts ("\n*****\n");
printf ("\nId de la Sucursal.....: %d \n", dato.idSucursal);
printf ("\nNombre de la Sucursal: .....: %s \n", dato.nombreSucursal);
printf ("\nNombre del Producto.....: %s \n ", dato.nombreProducto);
printf ("\nDeporte al que pertenece.....: %s \n ", dato.deporte);
printf ("\nStock del producto.....: %d \n", dato.stockProducto);
puts ("\n*****\n");
```

10 puntos

3. Codificar las estructuras y funciones que hagan falta para **recorrer la FILA y pasar los datos a un ARREGLO de LISTAS (simples) (ADL)**, clasificando los datos de la siguiente manera:

En cada celda del **arreglo** se ubica c/u de las sucursales de la empresa, de acuerdo a la siguiente struct:

```
typedef struct{
    int idSucursal;
    char nombreSucursal[25];
}stSucursal;
```

En cada celda también se encuentra una **lista** con todos los productos que tiene en stock cada sucursal, de acuerdo a la siguiente struct:

```
typedef struct{
    char nombreProducto[25];
    char deporte[25];
    int stockProducto;
}stProducto;
```

Para realizar esto, se deberá hacer lo siguiente, **modularizando** de acuerdo a la **responsabilidad de cada TDA**:

- a) Codificar las estructuras necesarias para manejar el arreglo y las listas.
- b) Recorrer la Fila extrayendo los datos de acuerdo al comportamiento que le corresponde al TDA.
- c) Por cada dato extraído, deberán copiarse los campos correspondientes para crear las variables de tipo stSucursal y stProducto.
- d) Buscar la sucursal a la cual corresponde el producto a cargar en el ADL.
- e) Agregar una nueva sucursal si ésta no existe.
- f) Agregar un nuevo producto a la sucursal que corresponda. **Los datos deben ser insertados en la forma en que resulte más EFICIENTE hacerlo** (al principio?, en orden?, al final?).

35 puntos

4. **Mostrar** todos los elementos de la **estructura compuesta**, permitiendo visualizar cada sucursal con sus productos en stock.

Modularizar de acuerdo a la **responsabilidad de cada TDA**:

10 puntos

5. Nuestro cliente desea poder buscar un tipo de producto por su nombre y deporte al cual corresponde, y averiguar el stock del mismo en la sucursal que elija (por ejemplo, averiguar cuántas calzas de running hay en la sucursal Guemes).

A tal fin, deberán realizarse las funciones que hagan falta, **modularizando** de acuerdo a la **responsabilidad de cada TDA**, para:

- buscar en el arreglo la sucursal elegida por el usuario del sistema
- luego, buscar en la sucursal **todos** los productos que coincidan con el nombre y deporte elegidos por el usuario, y **sumar el stock** de los mismos. **Esta función deberá ser RECURSIVA.**
- Se aclara que puede haber productos repetidos (ej: el producto calza está repetido porque hay calzas de running, calzas de ciclismo, etc, y también puede haber distintos modelos de calzas de running o de ciclismo).

15 puntos

6. Hacer una función **main()**

- Para hacer esto, cree las variables que considere necesarias e invoque las funciones (de forma directa o indirecta) como corresponde en cada caso.
- Muestre los resultados cada vez que sea necesario.

10 puntos)